UNCLASSIFIED

AD NUMBER

AD147553

NEW LIMITATION CHANGE

TO

Approved for public release, distribution unlimited

FROM

No Foreign

AUTHORITY

ONR ltr., 9 Nov 77

THIS PAGE IS UNCLASSIFIED

Fed
fil

57-1

FC
BAC

#31

UNIVERSITY OF CALIFORNIA
LOS ANGELES

February 24, 1955

Management Sciences Research Project

Discussion Paper No. 51

PRIORITY FUNCTION METHODS FOR JOB-LOT SCHEDULING

by

Ross T. Nelson

Prepared while the author was under contract to Logistics Branch, Office of Naval Research, and employed on the Management Sciences Research Project, University of California, Los Angeles.

INTRODUCTION

The research reported in this paper was directed toward the development of *computationally feasible* methods for obtaining *approximations* to the solutions of certain *job-lot* scheduling problems.

Linear programming is valuable for describing and solving certain production planning and programming problems, but its usefulness is limited to situations wherein the assumption that production quantities are completely divisible is not too strongly contrary to reality. In particular, if there are substantial setup time requirements associated with the production activities, then linear programming does not seem applicable; we wish to consider such situations, in which the production orders or jobs are processed under the following conditions:

1. Each product has a required sequence of operations which must be performed by certain machines (or by certain types of machine).
2. Certain production orders are required to be processed as job-lots; no machine can work on two lots at one time, and no lot can be on two machines at one time.
3. There is a substantial expense and/or time-loss associated with setting up a machine to perform a given operation, so that "splitting" job-lots is expensive (we shall actually permit *no* lot-splitting in the particular types of problems to be considered below).

It seems very unlikely that exact analytic methods for solving problems of scheduling under such conditions will be developed in the near future, and even less likely that a computationally feasible method will be obtained. Such a method would have to select the best among all possible programs (with all possible lot sizes and lot-splittings), with respect to some "objective function" or "optimizer." This optimizer, for the exactness of the solution to have any meaning, would have to include measures of all costs and profits--somehow taking into account the intangibles like "customer good will." Thus, even the construction of a suitable optimizer is a formidable problem.

In these notes, therefore, we restrict ourselves to less general problems. No analytic solutions have been obtained for these problems (except in a few very special cases), but it is felt that the methods presented below will be useful.

Section 1 provides a detailed description of the two problems considered in this paper. The problems are intended to be illustrative of two types of job-lot production situations, which present different goals for detailed scheduling.

Section 2 is summary of the research on these problems, and a brief outline of a scheduling method.

Section 3 provides a detailed description of the scheduling method introduced in Section 2.

Section 4 is concerned with sample problems to which the author has applied priority function methods. Certain conventional scheduling procedures have been applied to the same sample problems. The different sets of results are presented for purposes of comparison.

Possible modifications of the method of Section 3 are discussed in Section 5. In addition, suggestions for the operational use of priority function methods and suggestions for further research are presented.

SECTION 1

Two types of job-lot production scheduling problems will be considered in the following sections. Section 1A describes a manufacturing process in which certain orders are to be processed, with no concern for the completion-times of individual orders. This is representative of a production system in which customer orders are filled from a stockpile of finished products, or where sub-processing units produce an inventory of parts for subsequent production operations or assembly. The goals of detailed scheduling in applications of this type may include reduction of overtime, reduction of sub-contracting, reduction of labor force, increased machine utilization, increased production capacity for a given time period, etc. (in any case, the ultimate purpose of detailed scheduling is to provide management with as efficient a plan as possible for producing a given bill of goods). For definiteness, we shall assume in Problem 1A that an efficient production schedule is one with high machine tool utilization; i.e., the goal of the detailed scheduling in Problem 1A will be to make average machine utilization as high as possible (in our specific usage of "machine utilization," this is equivalent to minimizing overall production time).

Section 1B will be concerned with job-lot production systems wherein the due-dates of individual production orders are of importance. This is typical of job shops which produce directly to customer orders, and of production departments which produce for an intra-firm assembly schedules. In systems of this type, detailed scheduling procedures should be designed so that insofar as possible, production schedules will attain the individual production orders' due-dates. Problem 1B will thus be concerned with scheduling a bill of goods through the production process so as to "best" satisfy a set of due-dates for the individual production orders. We shall not precisely define the word "best," but shall generally think of trying to minimize the maximum tardiness or some other simple function of the differences between actual completion-times and due-dates.

Problems 1A and 1B are closely related, as will be indicated in more detail below.

There are, of course, many factors which could complicate the problems considered here. For example, the setup costs or times at each work center could be a function of the order of processing the jobs through that center, costs could be attached to the extra setups resulting from splitting lots, etc. This preliminary report does not consider such extended problems.

Description of the Problems.

The detailed job-lot production scheduling situation as considered here is the following: There are certain production orders (job-lots) to be scheduled through a plant containing certain machine tools (work centers). Each lot must be processed by certain machine tools in a certain technological order (routing), and for each lot there is a given expected processing time (standard time for the lot) on each machine tool needed for its processing.

A detailed production scheduling problem is one of assigning the given job-lots to the machine tools in such a way as to best satisfy some goal (such as maximum machine utilization), subject to the above restrictions, which we now restate more fully.

(1) No lot may be processed by more than one machine tool at one time.

(2) No machine tool may process more than one lot at one time.

(3) The lots must be processed by the required machine tools, each for the corresponding expected processing times, and in the required technological orderings.

(4) A lot must be processed as a unit; i.e., once started by a machine tool, its processing by that tool must be finished without delay, and the lot only becomes available for the next operation in its required sequence after its processing has been completed by the present tool.

We shall consider the problem of lot size determination as separate from the problem of scheduling. Specifically, we assume that the lot size of each production order is given, and (following requirement 4, above) that no lot-splitting is permitted. We also consider the routings to be unique.

1A - MACHINE UTILIZATION PROBLEM

Problem.

To schedule a given set of production lots so as to maximize average machine utilization (AMU), defined by

$$(1) \qquad \text{AMU} = \frac{\text{Total processing time}}{\text{Total processing time plus total idle time before the last required job-lot is completed}}$$

Discussion.

Let the production lots be designed by the integers 1, 2, ... , m , and the machine tools be designated by the integers 1, 2, ... , n . Let a_{ij} be the setup plus processing time for lot i on machine j ($a_{ij} = 0$ if the processing of lot i does not require machine j), and let T be the total time required by a given schedule until the completion of the last of the required job-lots:

$$(1a) \qquad \text{AMU} = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}}{n\,T} \quad .$$

The best possible value of the AMU will depend upon the relation between the machine tools available and the work required, and in a less direct way upon the other characteristics of the particular scheduling situation. If there are three machines, for example, and a bill of goods creates demand for only one machine, then the maximum possible AMU would be 1/3. This suggests that some other definition of AMU might for many purposes be more meaningful. However, we shall not explore this question here.

It will be observed that maximization of the AMU, as defined above, is equivalent o minimization of the total time until the completion of the last of the job-lots to be scheduled, and hence is equivalent to minimizing the maximum completion-time of the various production lots. This can be stated artificially as a due-date problem as follows: Assign to each lot the starting time of the entire schedule as a "due-date," and schedule to minimize the maximum tardiness. However, the due-date problem and the present problem seem to have substantially different characteristics in actual computation; the exact cause of this difference is not yet clear, but it justifies our continuing to consider the two problems separately, at least for the time being.

Example.

Suppose there are two jobs to be processed on two machine tools with operation times as follows:

	Mach. 1	Mach. 2
Job 1	4	1
Job 2	9	7

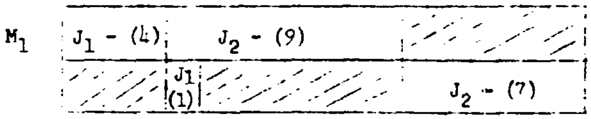
Suppose the technological restrictions for each job are that each job must be processed on M_1 before its processing may be on M_2 :

$$\text{Job 1: Machine 1} < \text{Machine 2} .$$

$$\text{Job 2: Machine 1} < \text{Machine 2} .$$

For this small problem there are only four possible detail schedules, the two best of which are shown below in Gantt chart form:

Schedule 1

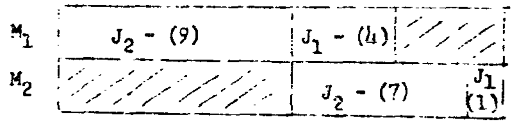


$\Sigma\Sigma a_{ij} = 21$

$T_1 = 20$ AMU $= \frac{21}{40} = .525$

$n = 2$

Schedule 2



$\Sigma\Sigma a_{ij} = 21$

$T_2 = 17$ AMU $= \frac{21}{34} = .618$

$n = 2$

For this problem the best possible AMU is .618. The second best schedule gives an AMU of .525 or only 85 percent of the best possible.

1B - DUE DATE PROBLEM

We consider now another type of problem wherein there is associated with each production lot a certain due-date.

Problem.

To schedule a given set of production lots so as to maximize some given function of the differences between actual completion times and due-dates for the various production lots.

Discussion.

The exact form of the function of the differences between actual completion times and due-dates will depend upon the particular application studied. In general, there are certain incremental costs associated with completing production before or after its due-date. These may include inventory charges, cost of assembling out of order, cost of holding other parts, loss of customer goodwill, etc. In a particular situation these cost elements might give rise to a cost function of the following type:

$$C_s = \sum_{i=1}^{m} (e_i x_i \Delta_{is}^E + l_i x_i \Delta_{is}^L)$$

where e_i is the incremental cost of producing one unit of commodity i one time unit ahead of its due date, x_i is the number of units of the commodity in the production lot, Δ_{is}^E is the number of units of time that the completion of lot i precedes its due date in a schedule s , so $e_i x_i \Delta_{is}^E$ expresses the incremental cost incurred by early production of lot i . Similarly, $l_i x_i \Delta_i^L$ is the incremental cost of late production of lot i (for a given lot i , either $\Delta_{is}^L = 0$ or $\Delta_{is}^E = 0$ or both).

The function C_s defined above is linear in Δ_{is}^E and Δ_{is}^L . In general, the costs of shifts in earliness or lateness might depend upon the degree of earliness or lateness, so that non-linear functions of the Δ's might arise; in fact, the costs associated with different job-lots might not even be separable. However, we are not primarily interested in the exact forms of these cost functions at present, the main purpose of this discussion being to suggest general properties relevant to the comparison of alternative production schedules.

In the sequel, we shall be concerned principally with cases in which penalties are associated only with lateness.

Example.

Suppose there are two job-lots to be processed on two machine tools with operation time requirements and due-dates as follows:

	Mach. 1	Mach. 2	
Job 1	4	1	Due-date = 6
Job 2	9	7	Due-date = 18

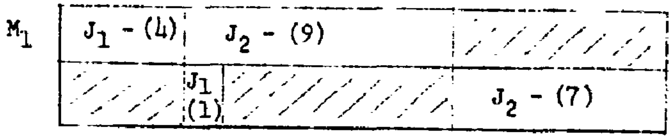
The technological orderings are:

Job 1: Machine 1 < Machine 2

Job 2: Machine 1 < Machine 2

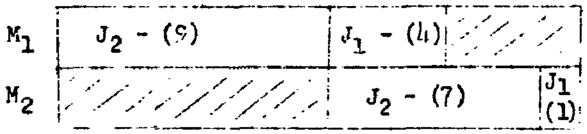
This is the same problem which was used to illustrate machine utilization in Section 1A, except that due dates are now given. If earliness is not costly, then the best schedule will be among the four according to which a machine is idle only when no work is available:

Schedule 1



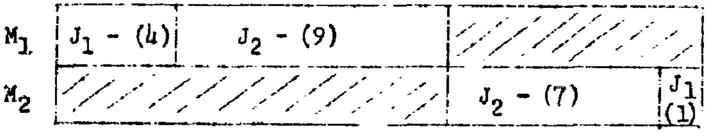
Completion Time Job 1: 5
Completion Time Job 2: 20
AMU = .525

Schedule 2



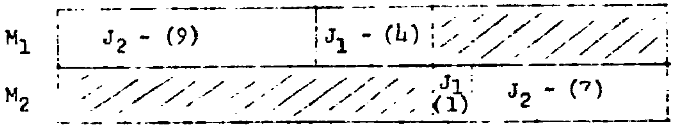
Completion Time Job 1: 17
Completion Time Job 2: 16
AMU = .618

Schedule 3



Completion Time Job 1: 21
Completion Time Job 2: 20
AMU = .500

Schedule 4



Completion Time Job 1: 14
Completion Time Job 2: 21
AMU = .500

It is clear that none of the schedules satisfies the requirement that each job be completed on time. If there are costs $l_1 s_1 = L_1$ and $l_2 x_2 = L_2$ associated with late completion of job-lots 1 and 2, the total cost due to deviations from due dates for the four alternative schedules are:

$2 L_2$, $11 L_1$,

$15 L_1 + 2 L_2$,

and $8 L_1$,

respectively. It is possible that either Schedule 1 or 2 be the most desirable, depending upon the coefficients. However, if the costs are reasonably homogeneous (as may be expected in practice) Schedule 1 will be best.

Summary.

Two simplified scheduling problems have been presented, the first concerned with maximization of machine tool utilization, the second concerned with attainment of pre-assigned due dates, and they have been solved by enumeration of all possible detail schedules.

Up to this point the concepts of machine utilization and due date satisfaction have been treated as separate, in order to emphasize two points of view. Maximum machine utilization has been used in representing a situation where it is desired only to schedule production so as to produce the required amounts as quickly as possible. Due date attainment has been used as a goal to represent a situation where it is desired to schedule production in such a way that individual job lots are completed at or close to specified times.

It must be recognized that in most applications it is desired to schedule so as to achieve a goal dependent upon intangibles as well as tangibles, and that even the precise statement of the objective scheduling is seemingly impossible. However, it appears that many real problems can reasonably be viewed as lying "between" the two extremes that have been discussed so that methods effective for these problems are a first step toward methods of comparatively wide applicability.

SECTION 2

The development of a detailed production schedule may be thought of as consisting of repeatedly answering the question: Which job-lot should be processed next on this machine tool? Any effective means for answering this question, whether by a dispatcher or foreman on the shop floor or by a high speed electronic computer, must utilize certain information. Consequently, the first step in this research was to try to list all factors which seem to be of basic importance in making sound decisions regarding job-lot assignments:

1. Availability of the job-lots for processing.
2. Final due date of each job-lot.
3. Current processing time for each job-lot.
4. Subsequent processing time for each job-lot.
5. Expected subsequent delays that each job-lot will encounter during the production process.

Remarks.

(1) A decision to process next on a machine tool a job which is not yet available for that machine tool will often result in idle machine time which may offset the advantages which suggested the decision. Thus, availability is of basic importance.

(2) The final due date is the time at which the job-lot is desired for subsequent processing, for inventory, for customer distribution, etc. It is apparent that these due dates should play an important role in job assignments.

(3) By "current processing time" is meant the expected (standard) operation time required to process the job-lot through the machine tool to which it is being assigned. Special consideration for this factor is suggested because the assignment of jobs requiring considerable processing time may block the progress of more urgent jobs requiring time on the same machine tool.

(4) By "subsequent processing time" is meant the expected operatime time for the job on all machine tools on which it must be processed, following the machine tool for which a decision is being made. This factor provides a measure of the relative progress of the various job-lots and is of obvious importance.

(5) By expected subsequent delays is meant the delays that the job is expected to encounter during the remainder of its processing. This is the least tangible of the factors to be considered, but one which may have a considerable effect on the decisions.

We next attempt to develop a logical decision-making procedure which will take into account the factors listed above. In order to facilitate discussion, we will adopt the following notation:

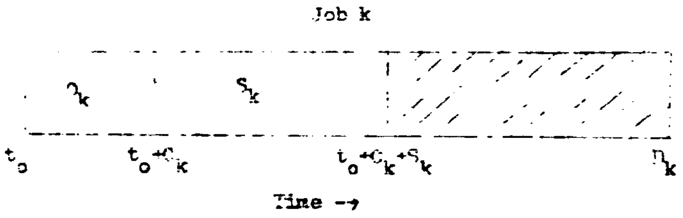
Denote due dates by D
" current processing times by O
" subsequent processing times by S
" expected delays by R .

Consider now a situation where a decision is to be made as to which job-lot will be next assigned to a specific machine tool. We will imagine that the machine tool has just completed a job-lot and is available for a new assignment. For the moment we will postpone consideration of job-lots that are to be processed on the machine tool but which are not yet available. This omission will not often result in large deviations from the optimum, except when long runs of items of widely varying urgencies are to be scheduled on machines whose capacity exceeds a plant's current needs.

For each of the available jobs the final due date (D), the current operation time (O), and the subsequent processing time (S) can be got from the manufacturing outline accompanying the work.

The manufacturing outline may also include information as to standard flow times based on historical experience of the plant. Such data might be useful to estimate the expected delay factor (R) which is to be considered in making decisions; but standard flow times reflect average shop conditions, rather than giving a measure of expected delays in view of the existing shop load. Hence we shall not use these data, but will suggest in Section 3 another way in which expected delays might be obtained.

If we denote by t_o the time at which the job assignment is to be made, we may represent graphically the information available from the manufacturing outline for each of the available jobs:



t_o - Current time

Q - Standard processing time on current operation

S - Standard processing time on all succeeding operations

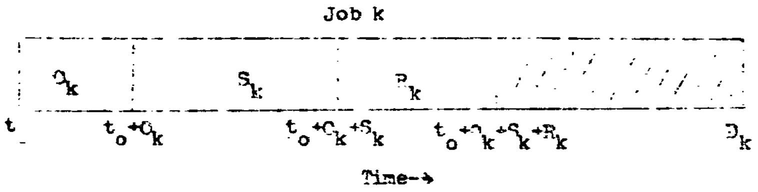
D - Final due date

The shaded area of the graph represents the time available before the final due-date, which will not be required for processing of the lot (i.e., it is a measure of the delays which the given lot may encounter as it progresses through the remaining part of its processing, and still be completed by the pre-assigned final due date). This time will be called the "slack". From the graph we see that the slack is given by:

$$A_k = D_k - [t_o + Q_k + S_k] .$$

In order to take the unknown delay factor into account, we may extend the graph to include an allowance for delays that the job-lot may be expected to encounter as it progresses through the remainder of its processing sequence. These delays may result from the following causes (scheduling delays discussed here do not include those due to machine down-time):

1. Delays due to work already assigned to machine tools when the job arrives at them.
2. Delays due to waiting for the processing of jobs with higher priority.



R_k - expected delays

Consideration of the expected delay factor R_k suggests another definition of "slack":

$$\Delta_k = D_k - t_o - O_k - S_k - R_k$$

In order to study the effects of job assignment decisions on the slack, let us assume that a particular job j from among the available jobs is assigned to a given machine at time t_o and investigate the results of such assignment.

(1) Job j will not be delayed at the machine; there will be no reduction in its slack.

(2) Each of the other available jobs will be delayed at machine M by a time equal to O_j ; the slack of each such job will be decreased by O_j units.

(3) The slack of each job which will become available at a time t_k $(t_o < t_k < t_o + O_j)$; will be decreased by $O_j - (t_k - t_o)$ units.

(4) The decision will have no immediate effect on jobs which will not become available for machine M before $t_o + O_j$; the slack for such jobs will remain unchanged.

The immediate effects of a specific decision may thus be measured by (1), (2), (3), and (4). In particular, the results may be summarized as follows:

1. No delay will be incurred on the assigned job j .
2. A delay equal to maximum $[0 , O_j - (t_k - t_o)]$ will be incurred for each other job k .

We refer to these delays as the "direct effects" of a decision. The adjective is needed because each decision that is made to assign a specific job-lot to a machine tool will not only delay the other job-lots currently at or approaching the machine tool, but will also have effects throughout the remainder of the production schedule. No useful method has been discovered for taking into account such indirect effects of job assignments in this decision making process (for an example of research in the combinatorial analysis of production scheduling, see Management Sciences Research Report No. 35 "Notes on Some Scheduling Problems," by James R. Jackson). Thus the work reported here is concerned with using information available in practical situations, and considering the "direct effects" of decisions, in order to develop approximating methods.

We have seen that the job slack as previously defined is a measure of the amount of unassigned time or the freedom for the particular job. The slack of a job will decrease each time the job is delayed. We have seen that the direct effects of assigning a particular job to a machine tool are to leave its slack unchanged and to decrease the slack of each job that is delayed by the decision. These observations emphasize that the slack times are not static quantities but are subject to continuous change. As a job is delayed more and more, due to other jobs being given priority over it, its slack will continue to decrease.

Now we suggest the following rule for use in job-lot scheduling when it is desired to satisfy final due dates:

> Whenever a machine tool becomes available, assign a job with smallest slack from among the jobs available at that time for processing on the machine.

Mathematically the rule may be stated as follows: Choose job j from among those available to minimize

$$\{ D_j - t_o - C_j - S_j - R_j \} .$$

We will call $\{ D_j - T_o - C_j - S_j - R_j \}$ the "priority function" associated with this rule. Detailed suggestions for use of this priority function are given in Section 3.

It is important to note here that this priority function is by no means the only one suggested by analysis of the problem. However, it has been chosen for initial development for the following reasons.

(1) It is a simple and reasonable function which is easily computed.

(2) Each decision is based on the current status of the various job-lots.

(3) At any time during the scheduling period new production orders or cancelled orders may be taken into account merely by adding or deleting the necessary data.

(4) The priority function involves the factors which were listed at the beginning of this section.

The attempt to estimate the delay factor (R_j) in the priority function (which is the only factor that is not known before any scheduling commences) leads to an iterative method for applying the formula; and we may hope that this procedure will lead to useful estimates of the delay factor. This procedure will be described in detail in Section 3.

The priority function may be easily revised, by merely eliminating the due-date factor D_j, to apply to the machine utilization type problem described in Section 1A.

SECTION 3

Computational Procedure. In this section we present a procedure for manual scheduling computations using the priority function method described in Section 2. Some of the alternative procedures which should be subjected to further study will be discussed at the end of the paper. However, the procedures considered would require relatively minor changes in the computational steps.

The computation is iterative; i.e., a schedule is developed using the priority function, then information given by the first schedule is used to develop a new schedule, information from the first two schedules is used to develop a third, etc.

The major steps for the computations are:

1. Make a table of job-lot data.
2. Development of the initial schedule.
    - A. Priority table.
    - B. Construction of schedule.
3. Iteration.
    - A. Delay table.
    - B. Priority table.
    - C. Construction of schedule.

1. Job-Lot Data Table. The job-lot table is a systematic tabulation of the pertinent data for each of the job-lots to be scheduled. All of the information in this table would come from the manufacturing outlines for the jobs.

Job-Lot Data Table

Job-Lot	Due Date	Standard Operation Times						Ready Time	Routing					

The job-lot column contains the coded job-lot identification number.

The due-date column contains the time at which the final operation on the job-lot should be completed.

The standard operation times columns contain the standard processing times (including setup and run time) for the job-lot at each operation listed in technological order.

The ready time column contains the coded time at which the job-lot will be ready for processing on its first operation.

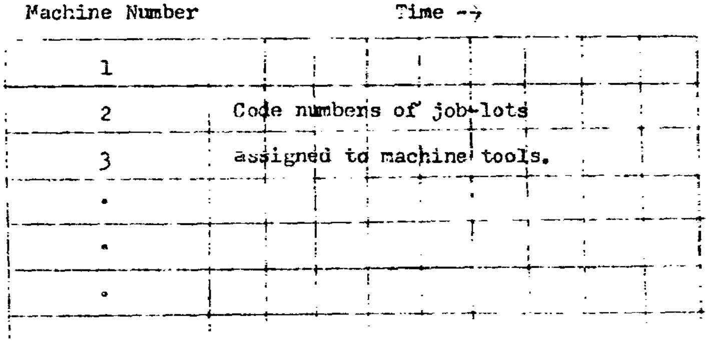
The routing columns contain the coded machine numbers listed in technological order.

2A. Priority Table. The priority table is a listing of the priority, as computed from the priority function, for each job-lot at each of its operations.

The R_{ij} term represents the expected delays during subsequent processing. In computing the initial schedule since no information is available for estimating these delays, we set $R_{ij} = 0$. The R_{ij} term will be non-zero in computing priorities for iterative steps later in the computations, and other initial estimates than zero will be considered below.

2B. Construction of Schedule. This step consists of using the job-lot data table and the priority table in order to construct a schedule based on the rule whereby one always assigns a job-lot, from among those available, with the smallest priority number (we recall that from the definition of priority number in Section 2, the smallest priority number represents the job-lot with the "highest" priority). Graphically, the schedule would be developed in a Gantt type chart similar to the one below:

Gantt Chart



Steps 1, 2A, 2B conclude the development of an initial schedule. The schedule which has been constructed is now used to estimate the delays which the job-lots can be expected to meet due to the interference of competing job-lots.

Priority Table

Operation Job-Lot	1	2	3	4	.	.	.
1							
2		Priority		numbers			
3		computed		from			
4		priority		function.			
.							
.							
.							

A priority number P_{ij} for job-lot i at its j-th operation is computed as follows:

$$P_{ij} = [D_i - O_{ij} - S_{ij}]$$

where:

D_i is the due date for job-lot i .

O_{ij} is the standard operation time for job-lot i on its j-th operation.

S_{ij} is the sum of the standard operation times for job-lot i on all operations following the j-th operation.

We recall that the priority function given in Section 2 was:

$$P_{ij} = [D_i - t_o - O_{ij} - S_{ij} - R_{ij}] .$$

The term t_o represents the time of the scheduling decision. Since, at any given time, t_o is the same for all available jobs, its presence merely adds the same constant to all priority numbers, so it can be omitted without changing the relative priorities of competing job-lots.

3A. Delay Table. The delay table for the first iteration is a tabulation of the delays which each job-lot experienced at each operation in the initial schedule.

Delay Table

Operation Job-Lot	1	2	3	4	.	.	.
1							
2							
3		Delays from					
4	initial schedule.						
.							
.							
.							

The delay table entries r_{ij} for the j-th operation on lot i are the number of time units that job-lot i had to wait in the initial schedule between the time it was available for the j-th operation and the time the operation was actually started. These are obtained by counting the number of time units in the Gantt chart of Step 2B between the completion of the (j-1) -st operation on job-lot i and the start of the j-th operation.

3B. Priority Table. The priority table for the first iteration is identical to the original priority table, except that the priority numbers are computed by the formula:

$$P_{ij} = [D_i - O_{ij} - S_{ij} - R_{ij}] ,$$

which takes into account the delay information from 3A. The delay factor R_{ij} is the sum of the r_{ij} from the delay table for all operations following the j-th operation.

3C. Construction of a Schedule. The construction of the schedule on a Gantt chart using the new priority table and the job-lot data table is identical to the construction of the initial schedule.

Further Iterations. The computations for further iterations would follow the same steps as those given for the first iteration escept that the delay information would be based on any or all of the schedules already constructed, rather than just on the initial schedule. For instance, the actual delays of the preceding schedule might always be used.

Example.

In order to illustrate the steps 1 through 4, we will compute a small-scale problem. We will suppose that the data from the manufacturing outlines for five job-lots to be scheduled have been transferred to the following job-lot data table:

Job-Lot Data Table

Job-Lot	Due Date	Standard Operation Times				Ready Time	Routing			
1	43	11	1	6	•	0	1	2	3	•
2	45	5	2	7	1	0	1	2	1	3
3	50	10	8	8	•	2	2	1	3	•
4	45	3	6	15	•	0	3	2	1	•
5	40	5	4	9	•	0	3	1	2	•

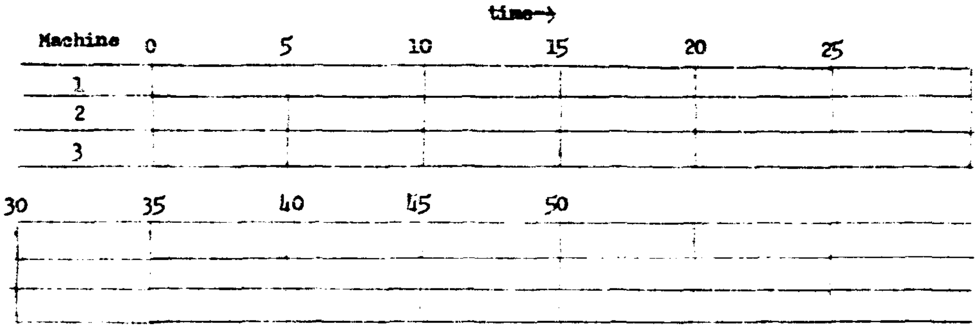
The priority numbers are next computed for each of the sixteen operations. For example, the priority for the first operation on Job-Lot 1 is $P_{ij} = [D_j - O_{ij} - S_{ij}] = [43 - 11 - (6+1)] = 25$. The priority table giving the results of these sixteen computations is given below:

Priority Table

Operation / Job-Lot	First	Second	Third	Fourth
1	25	36	37	•
2	30	35	37	44
3	24	34	42	•
4	21	24	30	•
5	22	27	31	•

It is convenient to add the priority table to the right side of the job-lot data table for ready reference.

We are now ready to construct the initial schedule in Gantt chart form. The blank Gantt chart for the three machines is drawn with unit time intervals:



(The time units are the same as the standard operation time units)

The scheduling rule is to assign to an available machine that one of the available jobs having the smallest priority number, so the mechanics of constructing the schedule on the Gantt chart from the job-lot data and priority tables are as follows:

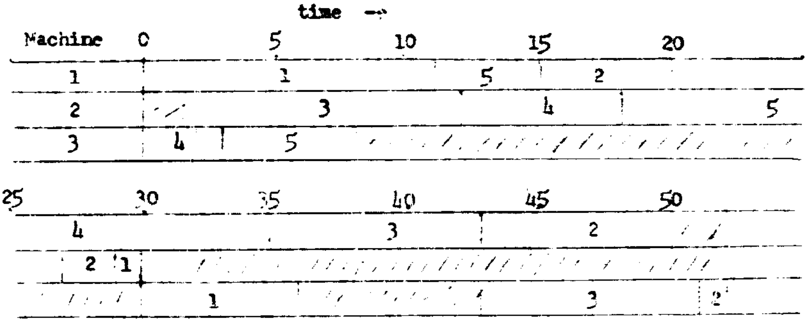
1. Starting at time $t = 0$, see which jobs are available for each machine tool that is not loaded. Look at the priority numbers and choose for each machine tool to be loaded, the job with lowest priority number.

2. Indicate the assignments on the Gantt chart by filling in for each machine being loaded the assigned job-lot numbers, for as many time units as the standard operation time for the job-lot on the machine.
3. After all possible assignments have been made proceed in time until one or more of the operations has been completed or until a job-lot becomes available for an unassigned machine. Cross out the data and priority number for completed operations and then assign job-lots to the available machines just as before.
4. Continue in time until all operations have been completed.

In the sample problem, all three machines are available for work at time = 0 . For machine 1, job-lots 1 and 2 are available because they both have a ready time of t = 0 , and both go to machine 1 first as indicated by the routing. The priority number for job-lot 1 on machine 1 is 25, for job-lot 2 it is 30. Thus, job-lot 1 is assigned to machine 1 at time t = 0 . The operation time is 11 units so a number 1 is filled in for the first 11 time units on machine 1 in the Gantt chart. For machine 2, there is nothing available at t = 0 so no assignment can be made. (Although job-lot 3 goes to machine 2 first, its ready time is t = 2). For machine 3, job-lots 4 and 5 are available. Job-lot 4 has the lower priority number and is assigned at t = 0 for 3 time units.

These initial assignments as well as all succeeding assignments are given on the following completed Gantt chart.

Initial Schedule - Gantt Chart



The delay table formed by measuring the delays in the initial schedule is given below:

Delay Table

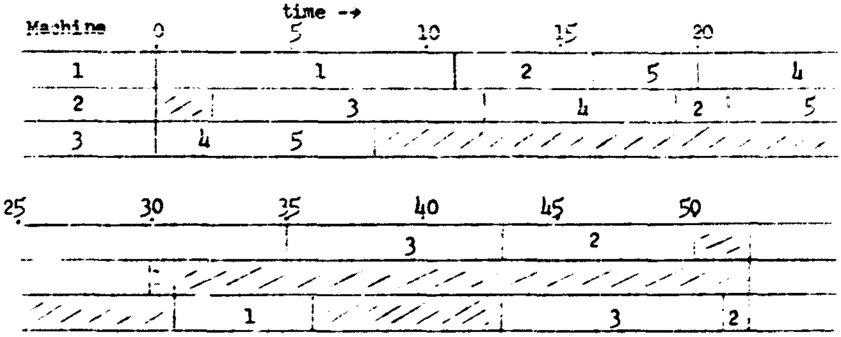
Operation / Job-Lot	First	Second	Third	Fourth
1	0	18	0	•
2	15	7	14	1
3	0	23	0	•
4	0	9	2	•
5	3	3	3	•

A new priority table is now computed for the first iteration. The priorities are now computed by $P_{ij} = \lceil D_j - T_{ij} - R_{ij} \rceil$ where R_{ij} is the sum of the delay table entries for operations succeeding the j-th on job-lot i . For example, the new priority of job-lot 1 on its first operation is $\lceil 43 - 11 - (6+1) - (18+0) \rceil =$ 7 . The complete new priority table is:

Operation / Job-Lot	First	Second	Third	Fourth
1	7	36	37	•
2	8	20	36	44
3	1	34	42	•
4	10	22	30	•
5	16	24	31	•

The Gantt chart for the first iteration using the same procedure as before with the new priority numbers is as follows:

First Iteration - Gantt Chart



Further iterations would follow the same steps. This problem has been used merely to illustrate the computational procedure, and we will not analyze the results. In the next section two larger problems which have been used for preliminary research will be discussed and analyzed.

SECTION 4

This section includes some of the research computations performed for two synthetic problems. The two particular problems were chosen because they offered an opportunity to evaluate the results of the priority function scheduling. The first problem, taken from Alford and Bangs Handbook (p. 119), has 6 jobs and 9 machines, and is concerned with machine utilization. It was chosen as a research problem for priority function scheduling because the schedule so obtained could be compared with that given in the handbook. The second problem, with 17 job-lots and 6 machines, was specially constructed so that an optimum solution was known, so that the priority function results can be compared with an exact solution. The second problem will be discussed both as a machine-utilization type problem and as a due-date type problem.

Problem 1. A machine utilization synthetic toy problem.

Job-Lot Data Table

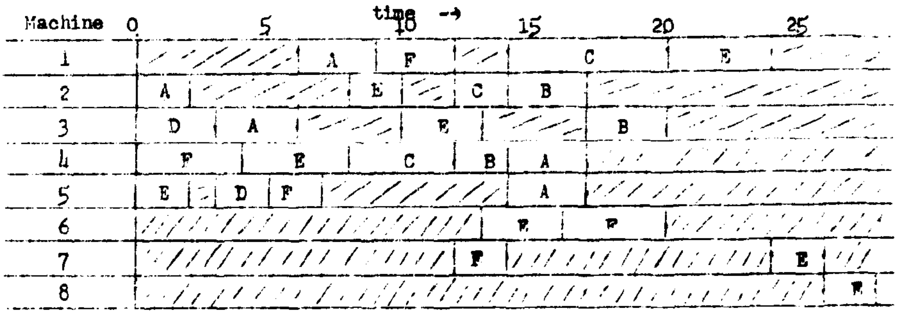
Job-Lot	Standard Operation Times								Ready Time	Routing							
A	2	3	3	3	3	•	•	•	0	2	3	1	4	5	•	•	•
B	2	3	3	•	•	•	•	•	0	4	2	3	•	•	•	•	•
C	4	2	6	•	•	•	•	•	0	4	2	1	•	•	•	•	•
D	3	2	•	•	•	•	•	•	0	3	5	•	•	•	•	•	•
E	2	4	2	3	3	4	2	2	0	5	4	2	3	6	1	7	8
F	4	2	3	2	4	•	•	•	0	4	5	1	7	6	•	•	•

Priority Table

Operation / Job-Lot	First	Second	Third	Fourth	Fifth	Sixth	Seventh	Eighth
A	14	12	9	6	3	•	•	•
B	8	6	3	•	•	•	•	•
C	12	8	6	•	•	•	•	•
D	5	2	•	•	•	•	•	•
E	22	20	16	14	11	8	4	2
F	15	11	9	6	4	•	•	•

(**Note:** Since this is a machine utilization problem, the due dates are omitted from the priority calculations; this amounts to replacing all due-dates by zero. The priority numbers become $P_{ij} = O_{ij} + S_{ij}$ and the largest priority number represents the highest priority. This is true because, if $D = 0$, then $\min [D - S - O_j] = \min [-S - O_j] = -\underline{\max} [S + O_j]$.)

Gantt Chart - Initial Schedule



Delay Table

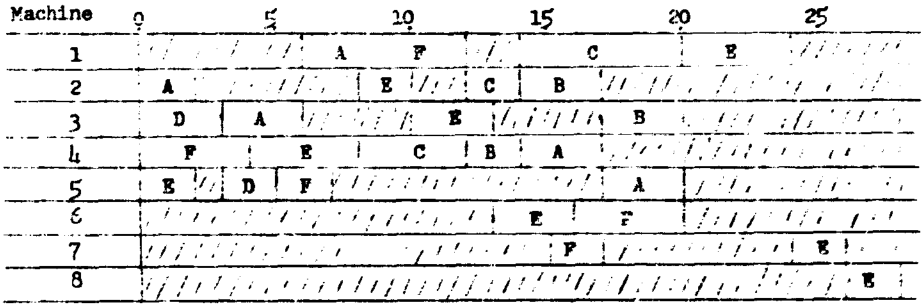
Operation / Job-Lot	First	Second	Third	Fourth	Fifth	Sixth	Seventh	Eighth
A	0	1	0	5	0	•	•	•
B	12	0	0	•	•	•	•	•
C	8	0	0	•	•	•	•	•
D	0	0	•	•	•	•	•	•
E	0	2	0	0	0	4	0	0
F	0	1	2	0	2	1	•	•

New Priority Table

Operation / Job-Lot	First	Second	Third	Fourth	Fifth	Sixth	Seventh	Eighth
A	20	17	14	6	3	•	•	•
B	8	6	3	•	•	•	•	•
C	12	8	6	•	•	•	•	•
D	5	2	•	•	•	•	•	•
E	28	24	20	18	15	8	4	2
F	20	15	11	8	4	•	•	•

Gantt Chart - First Iteration

(AMU = 34 percent)



In this case, the schedule resulting from the first iteration is identical with the initial schedule. It is clear that further iterations using the same method would not result in any change in the resulting schedule. The overall time required for processing the job-lots in the schedule given in the Production Handbook is [illegible] time units. The priority function solution above requires but 28 units. Thus, the machine utilization is increased considerably in this case by employment of the priority function method.

A number of slight modifications of the exact iterative procedure which has been described here, were employed on this problem. The modifications were of two types:

1. Slight variations in the priority function.
2. Changes in the way the delay information was used for iteration.

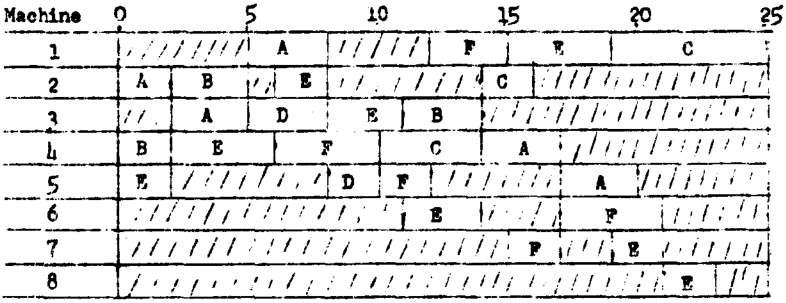
In all the cases tried, the resulting schedules were of overall length either 27 or 28 units. Thus, from these results nothing is indicated regarding the desirability of such modifications.

It is of interest to mention another method which was employed for this problem. We recall that we limited ourselves earlier to the set of available jobs at a machine when making job-lot assignments. However, for this small problem we tried a method for considering jobs not yet at the machine. When making a job-lot assignment, we considered the basic priority (from the priority table) for the available job-lots, and the priorities of job-lots not currently available but already assigned to their immediately proceding operation were adjusted by subtracting the number of time units until the job-lot was scheduled to arrive. When a machine completed one job, both available and non-available job-lots were compared and the job with smallest adjusted priority was assigned next (possibly resulting in machine idle time).

The initial computation and the first iteration for this extended method gave the same schedule, presented below for comparison with the previous results.

Gantt Chart - Non-Availability Method

(AMU = 38 percent)



The over-all processing time is 25 time units, a considerable improvement over the 28 time units required from the availability priority function schedule. This example indicates that a priority function method which takes non-available job-lots into account may be desirable. Further study and experimentation on simple synthetic examples is necessary in order to gain conclusive evidence regarding such methods.

Summary, Problem 1. Priority-function methods considerably improved the schedule given in the Production Handbook. Slight modifications in the priority functions used had little or no effect (which is at least partly due to the small size of the problem), but a substantial further improvement resulted from considering non-available jobs. The two priority function methods gave AMU's of 34 and 38 percent, while the Gantt chart in the Production Handbook indicated an AMU of only 28 percent.

Problem 2. Machine Utilization.

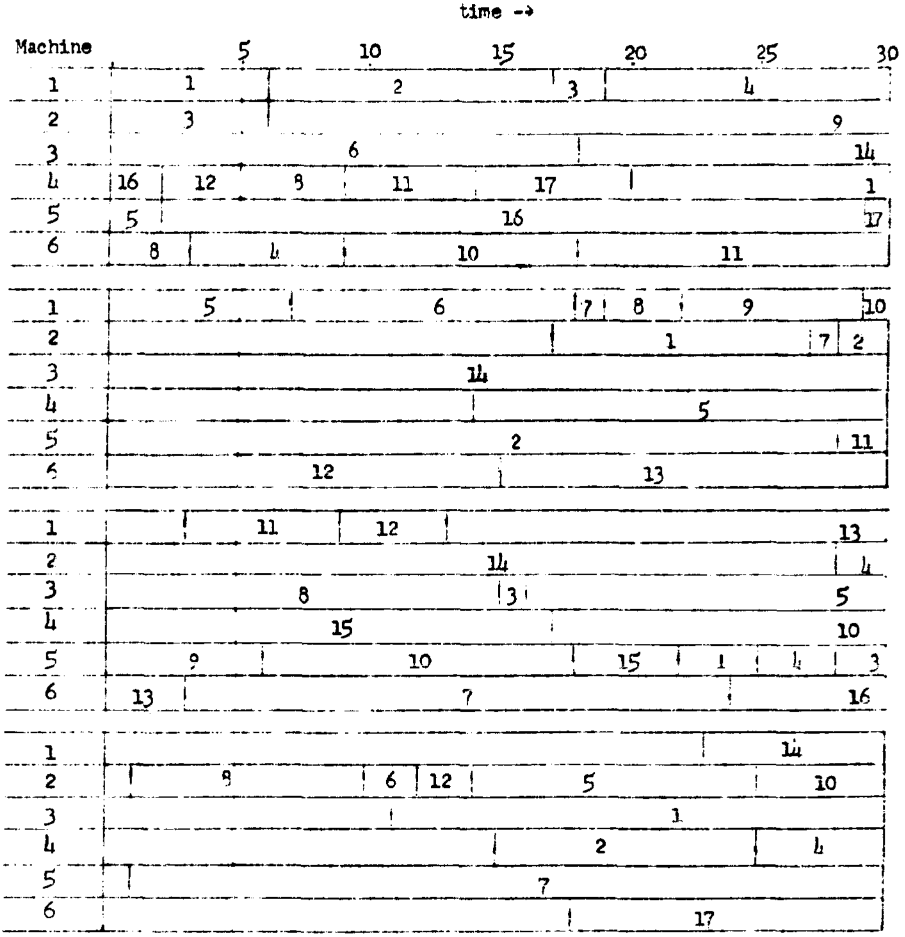
This problem was formulated by constructing a Gantt chart for which all machine tool are fully assigned for a given time period.

The job-lot data table and an optimum machine utilization schedule for the problem are given below:

Job-Lot Data Table

Job-Lot	Standard Operation Times	Routing
1	6 24 10 3 19	1 4 2 5 3
2	11 28 2 10	1 5 2 4
3	6 2 1 3	2 1 3 5
4	6 11 3 3 5	6 1 5 2 4
5	2 7 16 24 11	5 1 4 3 2
6	18 11 2	3 1 2
7	1 1 21 29	1 2 6 5
8	3 4 3 16 9	6 4 1 3 2
9	41 7 6	2 1 5
10	9 4 12 27 5	6 1 5 4 2
11	5 12 2 6	4 6 5 1
12	3 15 4 2	4 6 1 2
13	18 40	6 1
14	42 28 7	3 2 1
15	18 4	4 5
16	2 27 24	4 5 6
17	6 1 12	4 5 6

Gantt Chart - Optimum Machine Utilization



Obviously the machine utilization cannot be improved. There may be other processing sequences that are as good as the one above, but none could be better.

A number of priority functions were tried on this job-lot data, as a machine utilization problem. It would not be justified to draw conclusions from this one problem regarding the relative merits of alternate priority functions. The machine utilization for the initial schedule varied between 91 percent and 100 percent for the methods tried. The best results were obtained by starting with estimated equal delays at each machine (instead of initial assumptions $D_{ij} = 0$), such that the sum of the estimated delays for each job was equal to the job-lot slack (using as "due-date", a convenient value for all job-lots). A fixed priority system, based upon the total processing time for the job-lots, was applied to the problem in order to simulate a conventional rule for scheduling by priorities. The best and worst results in the initial schedule for the priority function methods used and the fixed priority number results were are follows:

Fixed Priority Number Based on Total Processing Time	- AMU	=	81	percent
Priority Function Method - No Initial Delay Estimates	- "	=	91	"
Priority Function Method - Initial Delay Estimates	- "	=	100	"

Problem 3. Due Date Satisfaction.

In order to transform the machine utilization version of Problem 2 into a due-date problem, the actual completion time of each job-lot from the original Gantt chart was used as the job-lot due date. Thus, the due-dates *can* be satisfied; but it seems reasonable that such satisfaction will not be easy. In the due date version of this problem, the same type of computations were used as in the machine utilization problem. The table below outlines the results that were obtained in the initial schedules. The AMU, which may still be of interest, is included.

Results of Initial Schedules

Scheduling Method	Total Tardiness All Job-Lots	Maximum Tardiness	AMU Percent
Fixed Priority Number System - Priority = Due Dates	212	60	67
Fixed Priority Number System - Priority = Job-Lot slack at first operation	203	61	73
Priority Function Method - No initial delay estimates	44	13	90
Priority Function Method - Initial delay estimates	29	16	96

In addition to the development of initial schedules using the various fixed priority number and priority function methods, a mass of computations were performed in order to compare various methods for iteration. The results of this work do not justify final conclusions, but they serve to indicate certain possibilities which should be the subject for further study on planned examples. This is discussed further in Section 5.

SECTION 5

Modifications of Priority Functions.

The problem of determining a particular priority function best suited to applications requires much more research. The particular priority function which would be best for a given problem may depend upon such features of the problem as the number of job-lots, number of machines, job-lot processing times, machine loads, technological orderings, due-dates, etc. In any event the value of a specific method will be determined by the results which the method will yield and by its cost. The cost factor makes it reasonable to conduct initial research on rather simple priority functions, to study the relative merits of simple methods on synthetic problems, and to obtain time and cost estimates for the computations required if these methods are to be applied in practice.

As examples of modified priority functions which might be the subject for experimentation on other synthetic problems, we suggest the following:

$$(1) \qquad \min_{j \in s} \left[D_j - S_j \right]$$

This form differs from the one previously described in that the immediate operation time O_j does not appear in the function. This suggestion is based on another simplified derivation of a logical criterion for minimizing "direct effects" of assignments.

$$(2A) \qquad \min_{j \in s} \left[D_j - O_j - S_j - N_j \right]$$

$$(2B) \qquad \min_{j \in s} \left[D_j - S_j - N_j \right]$$

These are variations of the previous functions, with the additional factor N_j, which is meant to reflect the number of operations to be done as well as the load on the associated machines. The factor M_j might be computed for example as follows:

Obtain an "average delay factor" m_i for each machine tool by

$$m_i = \frac{1}{2} \; \frac{\text{Total Processing Time on Machine } i}{\text{Number of Job-Lots using Machine } i} \text{ , and let}$$

$$M_j = \text{sum of } m_i\text{'s for subsequent operations.}$$

$$(3) \qquad \underset{j}{\text{Min}} \left[D_j - O_j - S_j + \Delta t_j \right] \text{, where } \Delta t = \text{time until job } j \text{ available.}$$

This form would be used to compare jobs at or approaching the machine. While there is evidence that such a modification may lead to significantly improved results, it seems reasonable first to conduct more studies on availability methods.

Modification of Iteration Procedure.

The iteration procedure includes: 1) the initial delay estimates; 2) the measurement of delays from the constructed schedules; and 3) the use of the delay information at one step to adjust the priorities for the succeeding step. The method described in this paper consisted of assuming zero delays to construct the initial schedule, and measuring the number of time units delay in the last schedule in order to adjust the priorities for the succeeding iteration.

Among the possible modifications of this iteration procedure are the various combinations of the following ways of handling (1) and (2).

(1) We may initially estimate the delays as:

a) Zero.

b) Equal for each operation on a job-lot; the assumed value for each operation might be $= \dfrac{\text{Total slack for job-lot}}{\text{Number of operations on job-lot}}$.

c) Equal for all jobs on one machine tool; the assumed value might be $\dfrac{1}{2} \; \dfrac{\text{Total time unit load for machine}}{\text{Number of job-lots using machine}}$.

d) A function of both the job-lot and the machine, combining the ideas of (b) and (c).

(2) The delay information may be measured by:

a) The delays in immediately preceding schedule.

b) The average delay in all preceding schedules.

c) The average in all preceding schedules and the initial estimates of delays as chosen in (1).

d) Some weighted average of the delays in preceding schedules and the initial estimates.

The interesting factors in evaluating these or other iteration procedures would be the effect on the resulting schedules in terms of the scheduling goal and the convergence of the iteration procedure. A number of these iteration procedures were employed for sample problem 2. Indications were that iteration procedures involving averages tended to converge in a short number of steps to a recurrent schedule. Other methods did not always converge; rather, continued iteration yielded a number of different schedules of approximately equal worth. However, the non-converging iteration methods often seemed superior to the convergent methods. We can only conclude that the comparison of iteration procedures requires further experimentation.

Operational Use of Priority Function Methods.

In thinking of potential applications of any scheduling methods, it is necessary to realize that the actual day-to-day shop occurrences such as machine breakdowns, operation times, etc., can never be exactly predicted. It is possible, at best, to include in the scheduling computations data which reflects predicted distributions of such parameters (our methods are concerned only with expected values). It is clear that the best obtainable schedule based on the expected values of the parameters may not be best for the actual conditions which occur during the processing in the shop. Hence, it seems reasonable to apply priority function methods in a sufficiently flexible way to allow for the deviation of actual conditions from expected conditions. This might be accomplished by computing a schedule based on expected operation times, expected orders, expected machine breakdowns, etc. and making up from it a "master priority list" for each machine type. The master priority list would provide the foremen with a guide from which they would deviate only if actual shop conditions dictated the need for a deviation (e.g., if the job to be processed next according to the master list is not available).

Another promising way to send computed data to the shop would be in the form of the priorities themselves (printed on the work tickets corresponding to each operation on each job, or printed in lists for each machine). When a machine finished one job, the foreman would select the highest-priority job among those actually available. Of course, experimentation is necessary to evaluate such procedures.

Areas for Further Research.

A study is being conducted by the Management Sciences Research Project to determing the feasibility of using existing electronic computing equipment for priority function scheduling. The purposes of the study are to estimate the time and cost which might be required for large scale practical scheduling problems and to program the method for an available computer in order to perform experimental work on larger synthetic problems.

The construction of problems which will give an efficient indication of the value of alternate methods is in itself a major difficulty. This is true because optimum attainable schedule can only be obtained for very special types of problems. However, there is the danger that if the experiments are restricted to such special problems, then conclusions regarding the relative merits of various methods cannot be confidently extended.

In addition to investigations regarding the priority function methodology and associated computational requirements, there are numerous research possibilities connected with special questions which may arise in practical situations. Problems of this type include:

1. How would partial technological orderings (alternate routings) in place of complete orderings affect the scheduling method?
2. Can lot size determination be associated with the detail scheduling method, i.e., can scheduling methods be devised to consider variable lot size possibilities? (See Management Sciences Research Project Research Report No. 28 "Job Shop Scheduling - An Application of Linear Programming," which suggests a linear programming method for lot size determination when the schedule is essentially predetermined.)
3. How can statistical parameters representing occurences such as machine breakdown best be included in the scheduling method?
4. What factors determine how long a time period should be scheduled at one time?

5. Will consideration of transportation operations suggest any basic change in the scheduling methods.
6. What other simple optimizers might be used to describe possible management goals in detailed scheduling.

The studies in these areas have only begun. However, it is felt that this discussion paper may be of use to others with research interest in production scheduling.